

In the Claims:

Please amend the claims as follows:

1. (Currently amended) A method for providing parallel execution of computing tasks in a heterogeneous computing environment having a plurality of computing hosts, comprising:

partitioning a computing task into small tasks;

assigning the small tasks to mobile agents;

monitoring the computing hosts for detecting an indication that migration needs to be initiated;

determining available computing hosts in the heterogeneous computing environment;

automatically transferring the said mobile agents to the determined available computing hosts responsive to a detected indication that migration needs to be initiated;

executing the said mobile agents at the available computing hosts using execution code provided to the said mobile agents from a central server; and

maintaining, at a second computing host at which none of the said mobile agents are executing, stack trace and state information about each of the said mobile agents to allow one or more of the said mobile agents to be reconstructed at an alternate computing host using the said stack trace and state information.

2. (Currently amended). The method of claim 1, wherein said indication ~~the step of determining available computing hosts further comprises~~ an indication of determining network latency.

3. (Previous presented) The method of claim 1, further comprising transferring execution code from the central server to virtual machines at the available computing hosts.

4. (Original) The method of claim 1, wherein the step of executing the mobile agents is performed in virtual machines at the computing hosts without modification of the virtual machines.

5. (Original) The method of claim 1, wherein one of the small tasks is assigned to one of the mobile agents.

6. (Original) The method of claim 3, wherein the step of transferring execution code is performed in response to a request by a mobile agent.

7. (Original) The method of claim 6, wherein the step of transferring execution code is performed by a web server.

8. (Original) The method of claim 1 further comprising monitoring execution of the mobile agents at the available computing hosts.

9. (Previous presented) The method of claim 1 further comprising allowing the mobile agents to collaborate over the heterogeneous computing environment as the mobile agents execute at the available computing hosts.

10. (Original) The method of claim 1, wherein the mobile agents are executed in real time.

11. (Currently amended) A method for providing parallel computing using mobile agents comprising:

assigning a computing task to one or more mobile agents;

monitoring the computing hosts for detecting an indication that migration needs to be initiated;

transferring the one or more mobile agents to one or more available computing hosts responsive to a detected indication that migration needs to be initiated;

transferring execution code from a central server to the one or more available computing hosts;

executing the one or more mobile agents at the one or more available computing hosts using the execution code; and

storing stack trace and real-time state information about the one or more mobile agents at a first alternate computing host on which none of the one or more mobile agents are executing to allow the one or more mobile agents to be reconstructed at a second alternate computing host.

12. (Currently amended) The method of claim 11 wherein a detected indication is an indication of network latency, said method further comprising, prior to transferring the mobile agents, the ~~steps~~ step of:

~~determining network latency affecting transmission of data between computing hosts; and~~

halting transferring of the mobile agents if said detected network latency exceeds a threshold.

13. (Currently amended) The method of claim 12 comprising, ~~prior to transferring the mobile agents, the steps of:~~

~~monitoring utilization of the computing hosts; and~~

~~halting transferring of the mobile agents if utilization of computing hosts~~ fails to exceeds a predetermined threshold.

14. (Currently amended) A method for migrating a software application running in a virtual machine from a primary host to a secondary host comprising:

constructing an application using a plurality of mobile agents;

transferring the plurality of mobile agents to a first computing host;

executing the plurality of mobile agents at ~~the~~ said first computing host;

maintaining stack trace and state information about each of the plurality of mobile agents at a second computing host on which none of the plurality of mobile agents are executing;

detecting an indication to migrate the application, wherein said indication comprises one of the group consisting of: network latency, hostile attack, hacking, network failure or computer hardware failure; and

in response to ~~the~~ a detected indication, automatically migrating the application in its entirety from ~~the~~ said first computing host to a third computing host without modifying ~~a~~ the virtual machine at ~~the~~ said third computing host by reconstructing each of the plurality of mobile agents at ~~the~~ said third computing host using ~~the~~ said stack trace and state information.

15. (Currently amended) The method of claim 1 wherein said step of determining available computing hosts is performed in real time ~~14 wherein the indication comprises an indication of network latency.~~

16. (Currently amended) The method of claim ~~14~~1 wherein the indication comprises an indication of hostile attack.

17. (Currently amended) The method of claim ~~16~~1 wherein the indication comprises an indication of hostile attack ~~comprises hacking.~~

18. (Currently amended) The method of claim ~~14~~1 wherein the indication comprises an indication of network failure.

19. (Currently amended) The method of claim ~~14~~1 wherein the indication comprises an indication of computer hardware failure.

20. (Currently amended) The method of claim 14 further comprising resuming execution of the mobile agent at ~~the~~said third computing host at a point where execution was halted.

21. (Currently amended) The method of claim 20, wherein ~~the~~said stack trace and state information comprises information about an execution thread of the mobile agent as it existed at ~~the~~said first host prior to being transferred to ~~the~~said third computing host.

22. Cancelled

23. Cancelled

24. Cancelled

25. (Original) The method of claim 14 further comprising:
continuing monitoring for another indication to migrate the application;
continuing migrating the application to other hosts.

26. (Currently amended) A computer system ~~apparatus~~ for providing parallel execution of computing tasks in a heterogeneous computing environment comprising at least two computing hosts, said system comprising:

a dispatcher for partitioning the computing task into a plurality of small tasks and dispatching the small tasks;

mobile agents for receiving small tasks from the dispatcher;
means for monitoring execution of the mobile agents at the computing hosts;
means for detecting over-utilization of one of the computing hosts and for issuing
a warning when one of the computing hosts is over-utilized.

computing resources on a network including virtual machines for executing mobile agent software code;

means for transferring the mobile agents to the computing resources; and

means for automatically transferring execution code as well as stack trace and state information about each of the mobile agents responsive to detected over-utilization of one of the computing hosts from a central server to the computing resources, the computing resources receiving and executing one of the small tasks assigned to a mobile agent in the virtual machines using the execution code and the means for transferring execution code maintaining stack trace and state information about each of the mobile agents at a first ~~alternate~~-computing host where none of the mobile agents are executing to allow each of the mobile agents to be reconstructed at a second ~~alternate~~-computing host.

27. Cancelled

28. (Original) The apparatus of claim 26 wherein the central server comprises a web server.

29. (Original) The apparatus of claim 26 further comprising means for monitoring execution of the small tasks.

30. (Original) The apparatus of claim 26 further comprising collaboration means for allowing the mobile agents to communicate and share information in real time.

31. (Original) The apparatus of claim 26, wherein the mobile agents execute in real time.

32. (Original) The apparatus of claim 26, further comprising storage means for storing real time state information about the mobile agents as the mobile agents execute at the computing resources.

33. (Original) The apparatus of claim 26 further comprising:
means for monitoring execution of the mobile agents at the computing hosts; and
means for detecting over-utilization of one of the computing hosts and for issuing
a warning when one of the computing hosts is over-utilized.

34. (Currently amended) A method for providing realistic thread
migration which comprises:

instantiating a mobile agent thread at a first computing host;
processing ~~the~~ said mobile agent thread at ~~the~~ said first computing host;
storing stack trace and state information about ~~the~~ said mobile agent thread at a
second computing host at which ~~the~~ said mobile agent thread is not executing as ~~the~~ said
mobile agent thread executes at ~~the~~ said first computing host;
detecting an indication to migrate ~~the~~ said mobile agent thread; and
in response to ~~the~~ said indication;
stopping execution of ~~the~~ said mobile agent thread;
automatically transferring the execution code for ~~the~~ said mobile agent
thread from a central server to a third computing host; and—
automatically transferring ~~the~~ said stack trace and state information about
~~the~~ said mobile agent thread to a ~~the~~ said third computing host.

35. Cancelled

36. Cancelled

37. Cancelled

38. (Currently amended) The method of claim 34 further comprising:
receiving ~~the~~ said stack trace and state information about ~~the~~ said mobile agent
thread at said ~~the~~ third computing host;
reconstructing ~~the~~ said mobile agent thread at ~~the~~ said third computing host using
~~the~~ said stack trace and state information about ~~the~~ said mobile agent thread; and
continuing processing of ~~the~~ said mobile agent thread at ~~the~~ said third computing
host at a point at which execution of ~~the~~ said thread was stopped at ~~the~~ said first
computing host.

39. (Currently amended) The method of claim 38 wherein the step of automatically transferring ~~the-said~~ stack trace and state information about ~~the-said~~ mobile agent thread further comprises serializing the information about ~~the-said~~ mobile agent thread.

40. (Currently amended) An agent collaboration environment which comprises:

- a plurality of mobile agents;

- an agent debugger for storing stack trace and state information about each of the plurality of mobile agents at a computing host where none of the plurality of mobile agents are executing;

- a conference room for providing a virtual workspace for the mobile agents; and

- a registration subsystem for selectively assigning the plurality of mobile agents to the conference room,

wherein each of ~~the-said~~ plurality of agents can share data, information, and results of computations in the conference room.

41. (Currently amended) The agent collaboration environment of claim 40, wherein ~~the-said~~ conference room monitors and moderates communication between ~~the-said~~ plurality of mobile agents.

42. (Currently amended) A method for dynamically constructing and executing a computer application in a heterogeneous computing environment using mobile agents comprising:

- constructing a computer application using mobile agents;

- storing ~~the-said~~ mobile agents at a central server;

- dispatching ~~the-said~~ mobile agents from ~~the-said~~ central server to an available computing host on the heterogeneous computing environment using a dispatching component;—

- allowing ~~the-said~~ mobile agents to execute at ~~the-said~~ available computing host;

and

monitoring execution of each of the plurality of mobile agents and storing stack trace and state information about execution of ~~the~~ said mobile agents at a computing host where none of ~~the~~ said mobile agents are executing.

43. (Currently amended) The method of claim 42, further comprising monitoring CPU utilization and network latency prior to dispatching ~~the~~ said mobile agents.

44. (Currently amended) The method of claim 42, further comprising providing a collaboration environment for allowing ~~the~~ said mobile agents to exchange information in real time.

45. (Currently amended) The method of claim 42, wherein the step of allowing ~~the~~ said mobile agents to execute at the available computing hosts comprises allowing ~~the~~ said mobile agents to execute at ~~the~~ said available computing host without requiring prior installation of executable code at ~~the~~ said available computing host.

46. (Currently amended) The method of claim 1, further comprising reconstructing one or more of ~~the~~ said mobile agents at ~~the~~ said second computing host using ~~the~~ said stack trace and state information stored at ~~the~~ said first computing host.

47. (Currently amended) The method of claim 11, wherein the step of transferring ~~the~~ said one or more mobile agents to one or more available computing hosts comprises transferring data relating to ~~the~~ said one or more mobile agents to ~~the~~ said one or more mobile computing hosts.

48. (Currently amended) The method of claim 26, wherein ~~the~~ said means for transferring ~~the~~ said mobile agents to ~~the~~ said computing resources comprises means for transferring data relating to ~~the~~ said mobile agents to ~~the~~ said computing resources.

49. (Currently amended) The apparatus of claim 28, wherein the web server is located at a first organization.

50. (Currently amended) The apparatus of claim 49, further comprising an agent execution environment located at a second organization.

51. (Currently amended) The apparatus of claim 50, further comprising firewalls at ~~the~~said first and second organizations to allow execution code for ~~the~~said mobile agents to be exchanged between ~~the~~said first and second organizations.

52. (New) The method of claim 34 wherein the steps are carried out without the need to explicitly write any additional software code to either initiate, manage, or facilitate the transfer of a mobile agent during said process of thread migration.